DAA/Marshall NAS8-35979

ATMOSPHERIC PHENOMENA DATA

PROCESSING AND DISPLAY

ENGINEERING ANALYSIS, INC.

(NASA-CR-178448) ATMOSPHERIC PHENOMENA DATA PROCESSING AND DISPLAY (Engineering Analysis, Inc.) 22 p HC A02/MF A01 CSCL 04B

N86-16825

G3/47

Unclas 15646

EAI-TR-85-004A

ATMOSPHERIC PHENOMENA DATA

PROCESSING AND DISPLAY

Contract NAS8-35979

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Prepared by

Frank B. Tatom
Lynn R. Purser

July 30, 1985

ENGINEERING ANALYSIS, INC.

715 Arcadia Circle
Huntsville, Alabama 35801-5909
(205)533-9391

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

Under Contract NAS8-35979 with the System Dynamics Laboratory, Marshall Space Flight Center (MFSC), during the period from June 22, 1984 to June 21, 1985, Engineering Analysis, Inc. (EAI) has carried out several related efforts dealing with the analysis and display of data pertaining to various atmospheric phenomena. Technical efforts have been basically an extension of previous work [1].

Spectrometer data analysis, with refinements to the pre-existing programs, is discussed in Section 2. The discussion of a Monte Carlo simulation of a photon transfer problem is presented in Section 3. Raw data gathered in the field by a high-flying U2 aircraft is usually pulse code modulated (PCM). A description of techniques to reduce the data into a computer-amenable format is discussed in Section 4. Several utility programs and other work which has been carried out are discussed in Section 5, with references cited appearing in Section 6.

2. SPECTROMETER DATA ANALYSIS

Two broad band spectrometers (BBS), one for the visible (VIS) wave lengths and the other for the infrared (IR) wavelengths are carried aboard the U2 flights to gather data of the lightning spectrum above the cloud tops. The data stream from the spectrometer is pulse-code modulated and then recorded on magnetic tape. The broad band spectrometer pulse code modulated (BBSPCM) data must be made computer-amenable so that the researcher can perform analysis. Development of several programs along with enhancements to existing programs have been made to aid the researcher with support and ease in data reduction and analysis.

2.1 SPEC_PLOT PROGRAMS

SPEC_PLOT is the basic program that the researcher uses to analyze the BBSPCM spectral data. The program offers a variety of filtering techniques, background and D.C. offset removal procedures, an integration package, and a plotting routine, all of which the researcher can find very easy to apply to either the VIS orIR data. The goal has been to make the program user-friendly with accurate results.

2.1.1 Numerical Techniques

Two numerical filters are used to reduce high frequency noise components which are acquired by the signal from electronic noise and/or other sources. Both filters incorporate the "method of averaging" over several data points. This method acts as a low-pass filter on the data so that the higher frequency components are damped significantly. The equation used for the two filters is:

$$x_i' = 1/2\, x_i - 1/4[x_{i-j} + x_{i+j}]$$

where i = pixel number (2,3,..., 510)

j = pixel number increment (1,2)

x_i = unfiltered signal

x_i' = filtered signal

In addition to high-frequency noise, another component of the spectral signal that must be removed is the backscatter of light off of the cloud top, usually found in the daylight flights. The background removal technique uses a "running average" of previous frames to subtract from the present data frame. The main feature of this procedure is the presence of a memory buffer that keeps a history of the averages on a pixel-by-pixel (or wavelength-by-wavelength) basis. The equations used in the "running average" subroutine are

$$B_{i,k} = (1-\alpha)\, B_{i,k-1} + \alpha S^{c}_{i,k-1}$$

and

$$S_{i,k} = S_{i,k} - B_{i,k}$$

where

$i,k \in I^{+}$

$S_{i,k} \equiv$ signal from ith pixel of the kth frame

$B_{i,k} \equiv$ background estimate of the ith pixel for operation on the kth frame

$$S^{c}_{i,k} \equiv \begin{cases} S_{i,k} & (S_{i,k} \leq c) \\ c & (S_{i,k} > c) \end{cases}$$

$c \equiv$ limiting signal value

$\alpha \equiv$ constant set by user

The D.C. offset is determined by averaging the spectral data where the signal is very weak or where there is no signal response by the spectrometer. This average is then subtracted pixel by pixel to reduce the signal to the ground state.

Part of the analysis of a lightning event is the determination of the total energy under the spectral curve over a specified wave length interval. A subroutine to integrate spectral data was written and added to SPEC_PLOT. The subroutine allows the user to use default intervals of 50 nanometers or to define an interval of wavelength over which to integrate. The subroutine basically sums the spectral data points over the interval and outputs the results to a terminal.

2.1.2 Program Enhancements

An enhanced version of SPEC_PLOT was written to offer the user a smoother method to analyze data. The enhancements include user-friendly option choices, improved plotting procedures, and default parameters.

Default parameters were installed in SPEC_PLOT so that a "hands off" approach to data analysis can be done. The primary defaults setup includes the two numerical filters: D.C. offset removal and normalization factors applied. All the user must do is supply a spectral data file name and accept the default options.

Another user-friendly capability involves an easier way to access a desired file. The user can use either the job language Command Instruction (CI) generated files or File Manager (FMGR) generated files. The program allows the user to access multiple files without exiting the program. With the CI generated files the user can swap between the IR spectrum and the visible spectrum by simply answering the option questions.

The original SPEC_PLOT contained no way for the user to recall which filters were used, whether D.C. offset was removed, etc. unless the user kept his own handwritten record. A subroutine was added that keeps track of this information and writes the information of the plotted resuts. This makes it convenient to keep a record of BBSPCM spectral data already analyzed. Sample spectral data plots are provided in Figures 2-1, and 2-2.

2.2 D3_PLOT PROGRAMS (THREE-DIMENSIONAL TIME-RESOLVED PLOT OF SPECTROMETER DATA PROGRAM D3PLT)

The broad band spectrometer (BBS) data are stored on disk files in either 1024 words/record or 512 words/record. Each record or frame represents a scan corresponding to a 5-millisecond frame time resolution. Each pixel or data word is a functional value of the wavelength intensity integrated over 10.0 μsec. Every frame scan is PCM-encoded and then recorded on instrumentation tape recorders.

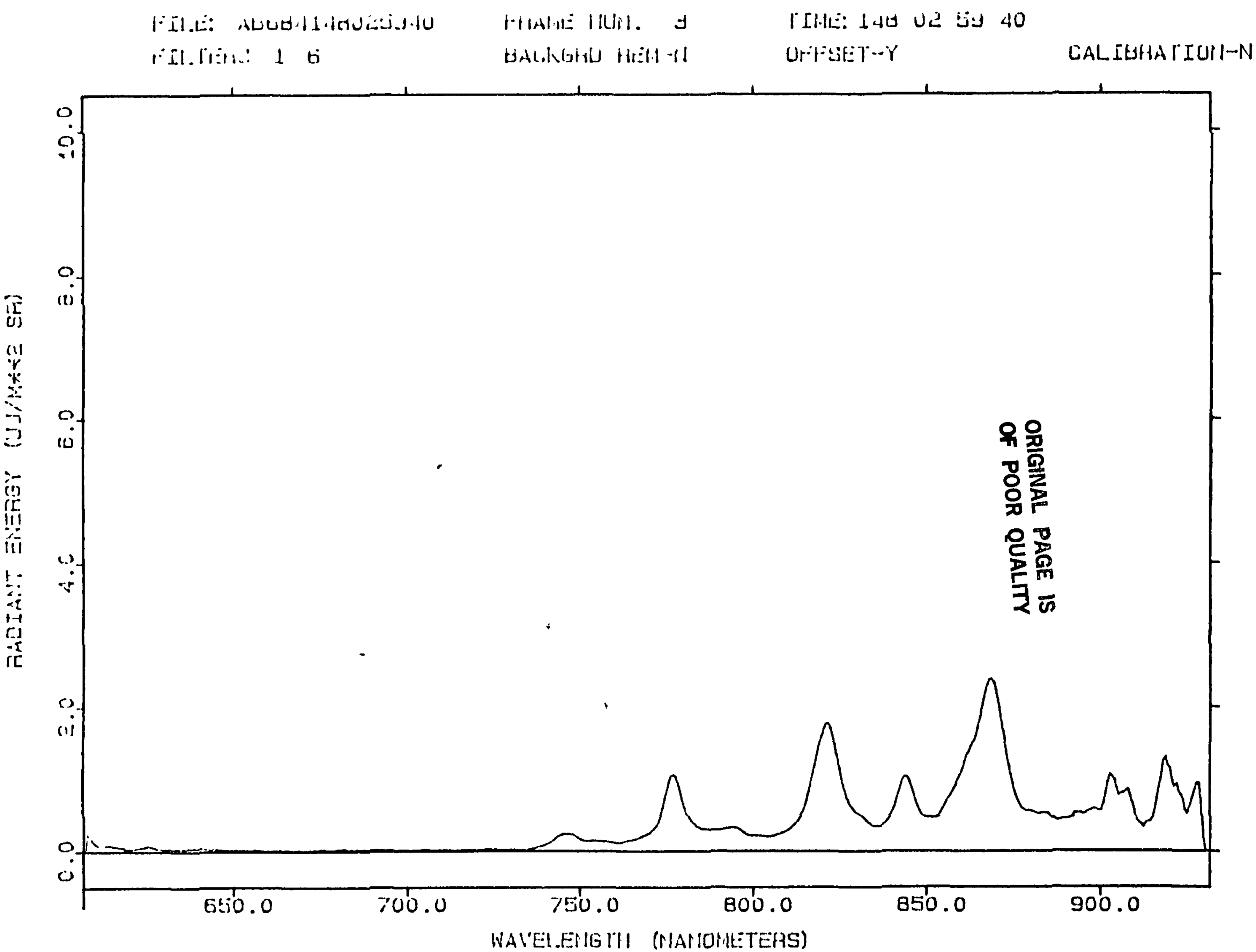

Figure 2-1. Radiant Energy Versus Wavelength from Broad Band Spectrometer Frame 3

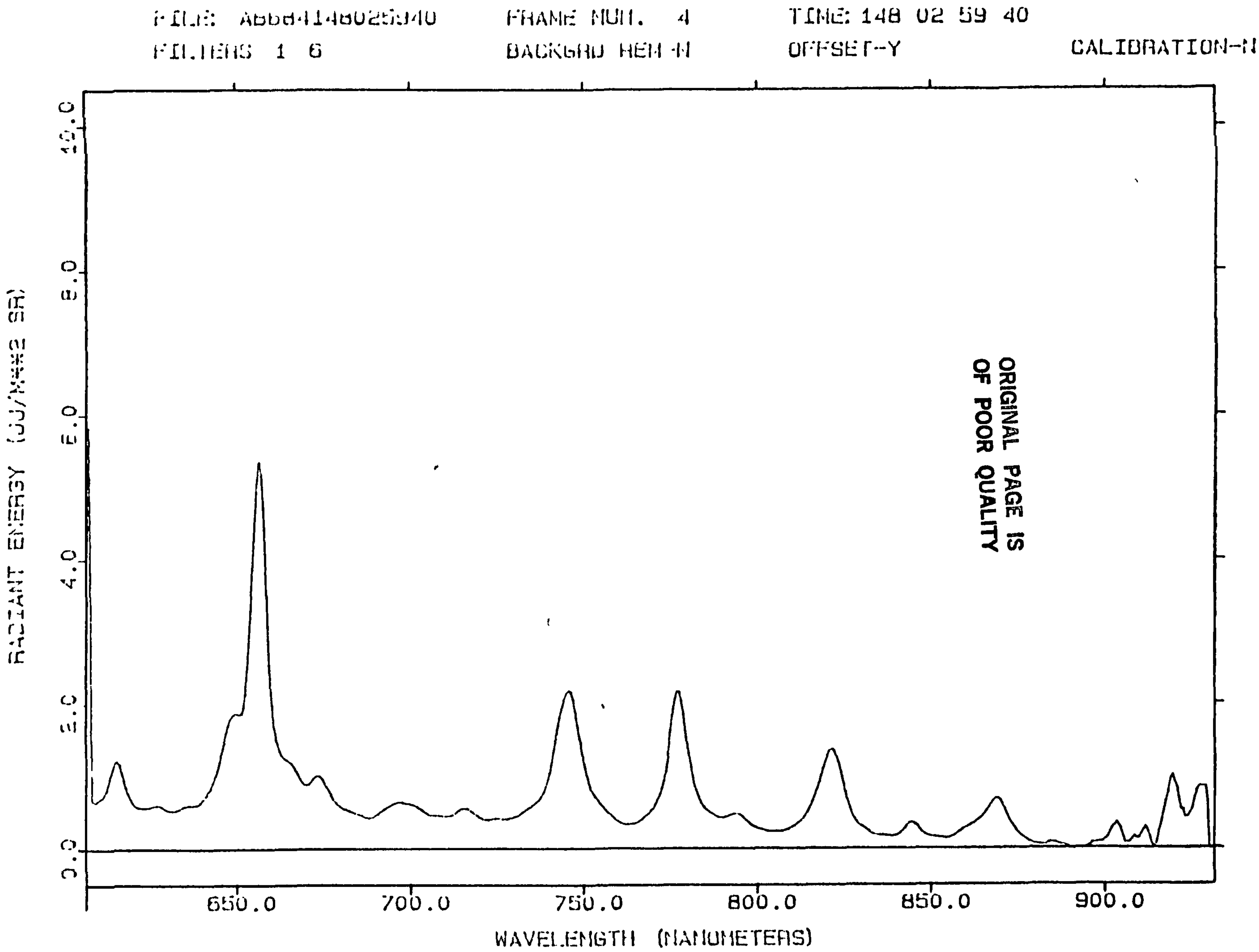

Figure 2-2. Radiant Energy Versus Wavelength from Broad Band Spectrometer Frame 4.

By means of the computer the frames of data are written to files stored on disk. The files contain approximately 50 frames of 512 words/record, which correspond to 512 pixel/frame. On this basis, there is a three-dimensional relationship: frame number (representing a five-millisecond increment in the time domain), pixel or word position (corresponding to wavelength), and intensity (value of the word.)

Based on the preceding arrangement a three-dimensional time-resolved plot routine, D3_PLOT, has been written. The axes X, Y, and Z are respectively wavelength, intensity, and frame number, oriented with X along the horizontal, Y along vertical, and Z into the page. The user is prompted for a default setup or the application of numerical filters, background removal, offset removal, and the angle in degrees in which the plane will be tilted. An algorithm converts the angle into the corresponding pixel translation, then translates the particular frame to achieve the three- dimensional effect. The translated frame is passed to a hidden-line removal routine, then plotted on an HP-9872 flatbed plotter.

D3_PLOT enables the user to get a"quick" look at the entire file to determine whether there are any interesting events which can be further analyzed. An example plot is shown Figure 2-3. The program also permits the user to study the structure of an event over several frames.

2.3 FOURIER PROGRAM

The FOURIER program, for use with the BBSPCM data, performs a discrete Fourier transform,

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk}$$

where

$$W = e^{-i2\pi/N}$$

$$n,N \in I^{+}$$

The program transforms the BBSPCM data, recorded in the intensity-vs-time domain, into the power-vs-frequency domain. The numerical results of the transform are plotted on a power-vs-frequency graph and the series coefficients are output to the printer. From these outputs the user can analyze

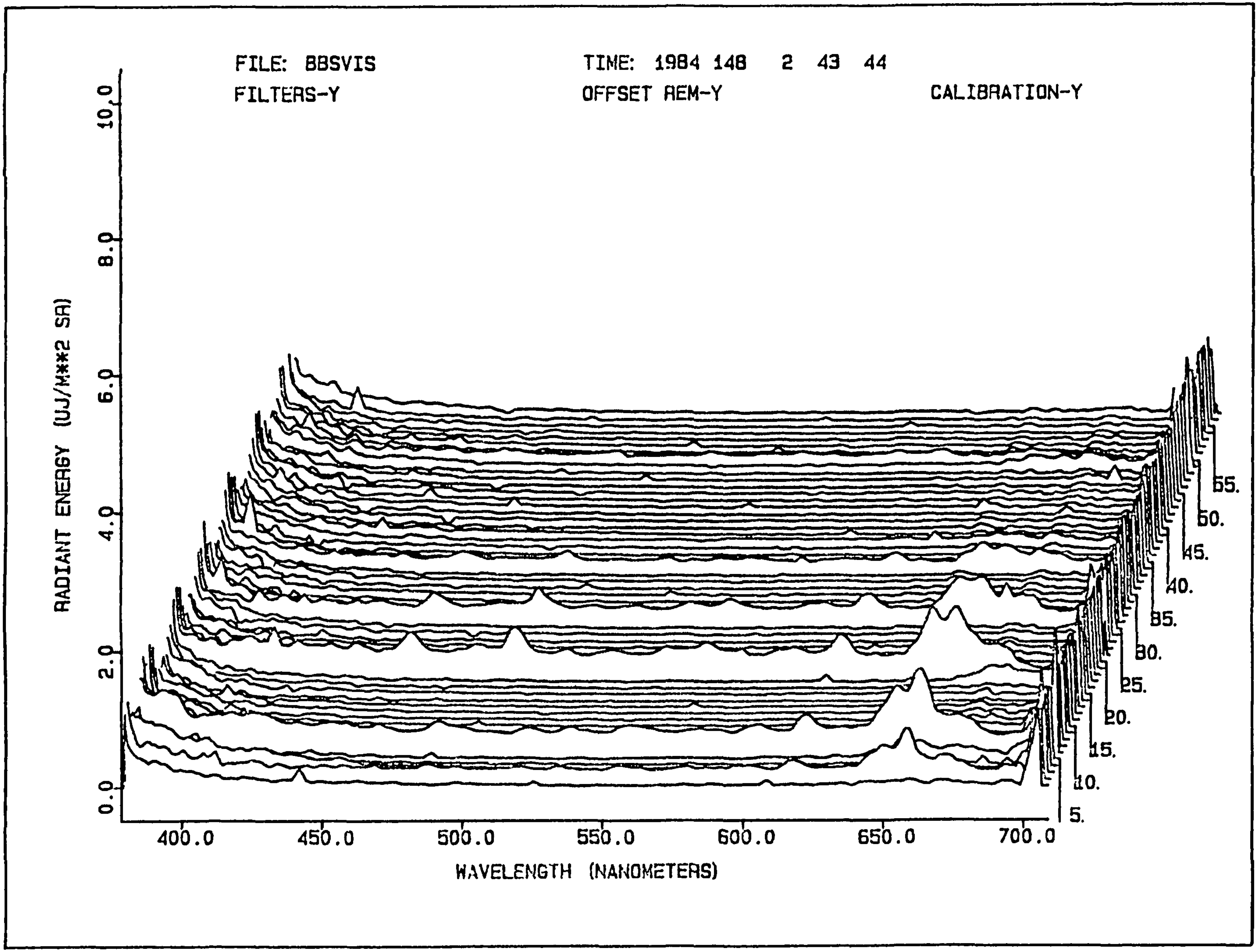

Figure 2-3. Sample Plot of Broad Band Spectrometer Data Generated by D3_PLOT

where the noise frequency is located and its power with respect to the signal. A filter subroutine in the program allows the user to apply various types of mathematical filters to eliminate or reduce the noise signal. The program then allows the inverse discrete transform,

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} (k)W^{nk}$$

where

$$W = e^{i2\pi/N}$$

to be performed on the frequency data, converting it back to the time series data, and plotting the results.

The FFT filtering technique produced results similar to those generated by the time-domain filtering process. Timing comparisons by MSFC personnel, however, indicated that the time-domain filtering process was faster. For this reason the FFT filtering technique has not been adopted as an analysis tool.

3. LIGHT PROGRAM

The original LIGHT program was written by Dr. E. P. Krider and L. W. Thompson, both of the University of Arizona [2], and has applications in the study of lightning above a cloud. The program, whose structure is described in subsection 3.1, uses the Monte Carlo method to simulate the transport of photons produced by some light source within a cubic cloud. In its original form LIGHT contained significant deficiencies, as discussed in subsections 3.2, 3.3, and 3.4. In an effort to provide a graphical display of the output of LIGHT, a special contouring routine was developed, as described in subsection 3.5.

3.1 PROGRAM STRUCTURE

The cubic cloud has dimensions x, y, z, and each face is divided into an 8x8 square array of bins. The bins keep a tally of the number of photons emitted within that particular area of the face. Other arrays within the program keep track of the azimuthal and zenith angles of emission. With the assumption that a light source is at the center of the cubic cloud, i.e. (x/2, y/2, z/2), in general each face has an equal probability of photon emission, with bins closer to the center of the face having a greater probability than bins near the edges or the corners.

3.2. ERRORS FOUND AND CORRECTED

The observation was made that several of the bins on each face received nearly all of the photons while other bins received none. This result implied that the number of calls to the random number generator of the HP-1000, A900 system exceeded the period of the generator. The correction of this problem is discussed in subsection 3.4.

Another discrepancy was that each face always showed a particular group of photon emissions in the corners of the faces. This particular discrepancy was in several outputs sent by Dr. Krider as well as in other runs made on different computers. By looking at the dump of the bin counts it was discerned that the counts were getting set in the output. This output was corrected and the problem of erroneous corner emissions was eliminated.

3.3. IMPROVED LOGIC

A flow chart of LIGHT was made to help in understanding of the logic, evaluation of the code, and possible restructuring to streamline the code. Several places in the program were identified as candidates for better logical coding.

In subroutine ANISOS there were three conditional checks to determine the quadrant of azmuthal angle (AZAG). The checks were necessary since AZAG was computed by first taking the absolute value of the quotient of the change in y (DYD) to the change in x (DXD), and then taking the ATAN of this argument. This angle was always in Quadrant I. By using three checks on the DYD and DXD the proper quandrant was determined. By using a double argument arctangent function (ATAN2(arg)) the proper quadrant can now be calculated without any additional tests.

Another logic deficiency was found in subroutine ESCAPE. In the process of determining through which face a photon escapes, the program checked all six faces. No matter if the first check proved positive, the program still did five more checks. The code was simplified using a case structure in FORTRAN. The case structures utilizes the IF<condition>THEN <operation>ELSE IF<condition>THEN<operation>ELSE IF. . . ENDIF construction. Therefore, if the first condition is met the subsequent checks are not made.

In conjunction with the two logic improvements noted, the entire program was restructured into a modular form for easier understanding of the logic. The overall result was a reduction in time of execution.

3.4 UPGRADE RANDOM NUMBER GENERATOR(RNG)

One problem in the LIGHT program on the NASA HP-1000, A900 system was the random number generator used in the Monte Carlo simulation. This RNG was inherent to the system, that is, the RNG was a callable function used in FORTRAN programs. The period of the generator, i.e., how long the sequence of random digits continues until it starts repeating, was determined to be 126,914 digits long. This number is far too short for application to photon transport.

Three new generators were programmed and run to determine which would give the statistically best sequence and longest period. The first generator was the linear congruential generator or LCG using the modular congruence relation [3],

$$x_{n+1} \equiv (ax_n + c) \pmod{m}$$

where

m - the modulus of the congruence ($m > 0$)

a - the multiplier ($0 \leq a < m$)

c - the increment ($0 \leq c < m$)

x_n - random number.

According to theory [5] the period of the LCG is $m = 2^p$ where p = wordsize of the host computer. Due to limitations of the HP 1000A. "p" was set equal to 23, which gave a RNG with a period of 2^{23} = 8,388,608 digits. However, even this period was not considered long enough for certain runs of LIGHT.

The second method incorporated coupled generators which has a period that is a combination of the individual periods [5]. For this method the LCG was used, coupled with the HP 1000A FORTRAN function RNG. The results extended the period to $2^{23} \times (126914)/2 = 5.32 \times 10^{11}$ random digits. This period was sufficient, but the generation time was excessive.

The third method, and the one that was selected to replace the original RNG, was the Generalized Feedback Shift Register (GFSR) [6]. This generator has an almost "unlimited" period without any increase to the wordsize of the computer. The generator uses the function

$$p(x) \equiv (x^p + x^q + 1) \pmod{2}$$

where the period is equal to $2^p - 1$.

The exponent, p, was chosen to be 98 so that the period of the random digits is $2^{98} - 1 \simeq 3 \times 10^{29}$, which is quite sufficient. The time of calculation for the FGSR was tested by generating a million random digits, and was found to be 27 seconds on the average. By comparison, the coupled generator required 1 minute, 6 seconds for the same calculation.

3.5 FACE_CON PROGRAM

A program was written (FACE_CON), using the EAI contouring subroutine, which allows the contouring of the photon distribution on the faces obtained from program LIGHT. As stated before, each face should show a symmetrical distribution about its center, given that the photon source is at the center of the cubical cloud. Several runs were made of LIGHT and the values of emmission count on each face were contoured. For runs individually under 1000 photons the results indicated that the distribution is not symmetrical but random.

For a run of 30,000 photons the contoured plots for the six faces can be seen in Figure 3-1. The plots are done on an 8 x 8 grid with the isocount lines starting with an initial value of 50 and increasing in increments of 15 to a final value of 275. Inspection of these plots reveals that the photon count is not precisely symmetrical about the center. This may be due, however, to the fact that 30,000 photons represents a relatively small population size.

ORIGINAL PAGE IS
OF POOR QUALITY

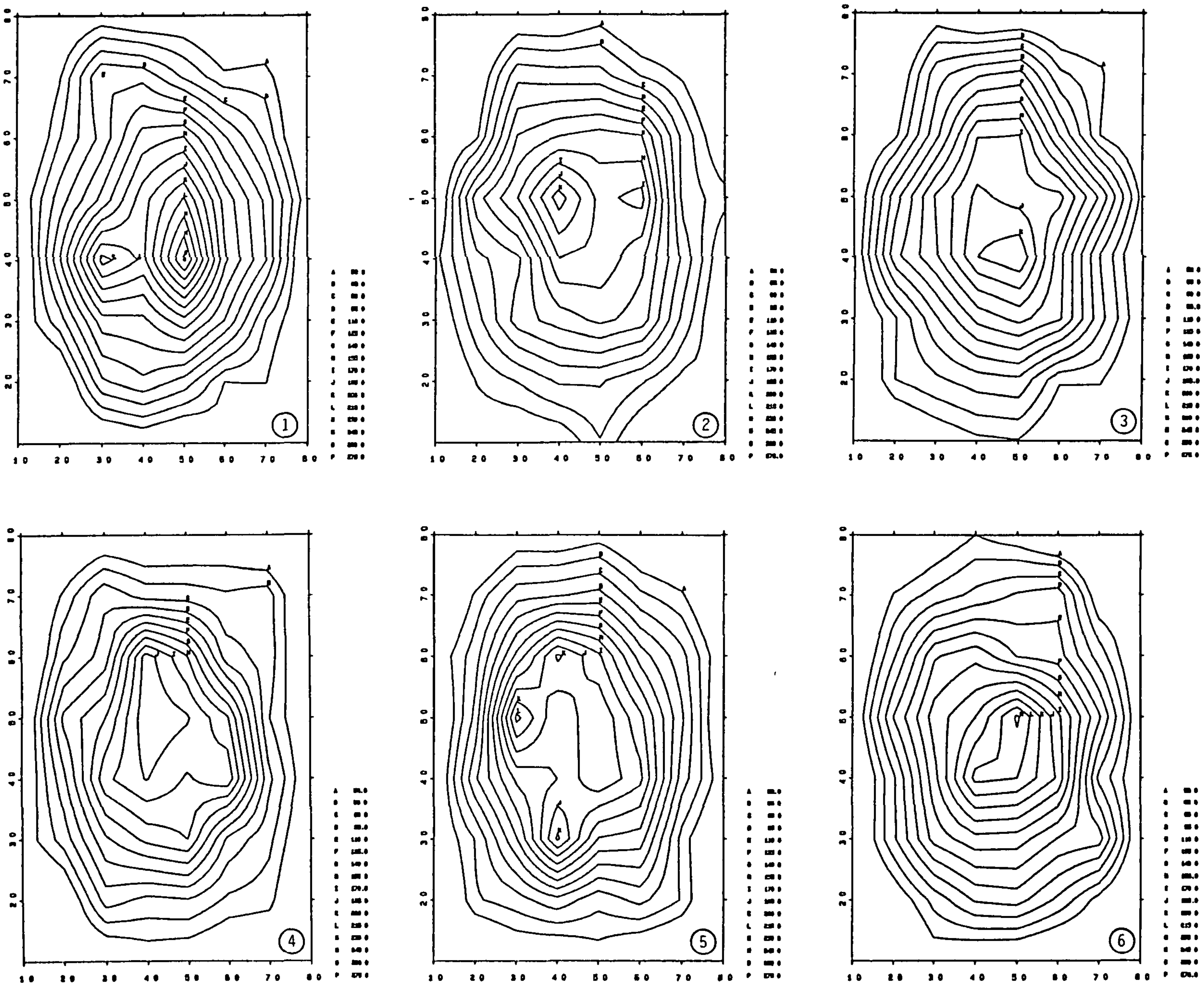

Figure 3-1. Photon Distribution on the Faces of a Cubical Cloud (30,000 photons)

4. PULSE CODE MODULATION (PCM)

During the U-2 field experiments Broad Band Spectrometer(BBS), High Resolution Spectrometer(HRS), and Optical Array Sensor(OAS) data were digitized and recorded using Pulse Code Modulation)PCM) on one channel of the analog tape. The data are later brought into the computer by the PCMIN program, discussed in subsection 4.1, and the BBS PCM data are processed further by the FRAME program, described in subsection 4.2.

4.1 PCMIN PROGRAM

The PCM data include the BBS, HRS, and OAS. PCM data are recorded on magnetic tape with the OAS signal read in first, followed by BBS and HRS. The PCMIN program controls a tape recorder, takes the raw PCM data off the tape, stores the data in a master disk file, then separates the master file into three files containing the OAS, BBS, HRS data respectively. Previously, for PCMIN to execute properly the tape recorder could only run at 1 7/8 inches per second (ips). The code was rewritten such that the tape recorder could run at 15 ips and perhaps faster. This increase in speed of the tape decreased the required acquisition time significantly.

In an effort to further improve the performance of PCMIN, certain changes to the internal logic were initiated. The data for PCMIN are brought into the computer in 2304 blocks of 128 words each of which are stored as a master file. Once the master file is filled, PCMIN releases control of the tape and starts the separation process of the master file into the OAS, BBS, and HRS files. The separation process previously took from three to four minutes. The original PCMIN made six checks at the head and tail of each of the data files to find an idle period, which indicated the beginning or the termination of a particular instrument's readout. These checks were not actually necessary because the number of data blocks of each of the instruments readout was known: The OAS file has 1024 blocks of 256 bytes each, and both BBS and HRS files have 512 blocks of 256 bytes each. By utilizing the number of blocks (or records of data) each instrument data file should contain, PCMIN has been revised so that the only check that is necessary is an initial check to find where the data starts within the master

file. Once the initial data has been found, large blocks of that particular instrument's data are read using the correct number of reads. As an example: The OAS file contains 1024 blocks of 256 bytes each. First, the point where the data starts in a block is determined. Next that block is read. Then eight reads of 128 blocks are made into a large buffer with enough memory for 2048 bytes. Finally, the buffer is written into disk file. Without the five additional checks the revised process takes from 9 to 15 seconds, representing a 92% reduction in computer time.

4.2 FRAME PROGRAM

The two spectrometer signals are interleaved by the onboard processor as they are read during the U2 flight. This method saves on storage and makes the process much faster. During the read process there are 1024 data words read followed by a 128-word blanking period. This blanking period must be removed to ensure accurate analysis of the BBSPCM spectral data.

The original FRAME program performed stringent "IF" tests on every block of 128 words to find the blanking period, sometimes with bad results. FRAME was rewritten so that the program finds the first word of the first blanking period, counts 128 words, then reads 8 blocks of 128 words each without any data checks, and fills a 1024-word buffer. The program checks the next block to find the blanking period and the initial data, and repeats the process until the end of data. This revision has decreased the processing time significantly and made the entire operation of BBSPCM spectral data analysis smoother and faster.

5. MISCELLANEOUS EFFORTS

5.1 PROGRAM TRANSFER

During November 1984 the HP-1000, A900 was installed, adding enhanced computing speed, more memory, and faster analysis of data. Many of the programs on the HP-1000F and the HP-1000L were transferred to the "A900" with total compatibility. The process of transferring the programs from the "F" to the "A900" involved transferring the programs via the Perkin-Elmer computer, because the two tape drives of the HP systems were of different BPI (byte per inch) ratio. The transfer from the "L" to the "A900" was done by tape and hard disc.

5.2 HRS (HIGH RESOLUTION SPECTROMETER) PROGRAM

D3_PLOT was modified to handle the HRSPCM data files. The modifications required rewriting sections that read in the datafiles BBSPCM data because the HRSPCM data are not interleaved and the records are 1024 bytes long.

6. REFERENCES CITED

1. Tatom, Frank B., Garst, Robert A., and Purser, Lynn R., "Processing and Display of Atmospheric Phenomena Data", NASA Final Report EAI-TR-84-003A, Engineering Analysis, Inc., Huntsville, Alabama, October 1984.

2. Thomas, L.W., and Krider, E.P., "The Effects of Clouds on the Light Produced by Lightning", Journal of the Atmospheric Sciences, Vol. 39, No. 9, September 1982.

3. Kusee, John, Elementary Abstract Algebra, D. C. Heath and Co., Boston, 1965.

4. Knuth, D.E., The Art of Computer Programming, Vo. 2: Seminumerical Algorithms, Addison-Westley, Reading, Mass., 1973.

5. Kenneth, W. Jr., and Gentle, James, Statistical Computing, Marcell-Dekker Publisher, Inc., 1979.

6. Lewis, T.G., Distribution Sampling for Computer Simulation, Lexington Books, 1975.